AFRL-AFOSR-UK-TR-2010-0016

Improving the Capacity of Language Recognition Systems to Handle Rare Languages Using Radio Broadcast Data

Lukas Burget

Brno University of Technology
Faculty of Information Technology
Bozetechova 2
Brno, Czech Republic 61266

EOARD GRANT 08-3066

January 2011

Final Report for 15 October 2008 to 15 December 2010

Distribution Statement A: Approved for public release distribution is unlimited.

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515 Box 14, APO AE 09421

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
16-01-2011

2. REPORT TYPE
Final Report

3. DATES COVERED *(From – To)*
15 October 2008 - 15 December 2010

4. TITLE AND SUBTITLE

Improving the Capacity of Language Recognition Systems to Handle Rare Languages Using Radio Broadcast Data

5a. CONTRACT NUMBER

FA8655-08-1-3066

5b. GRANT NUMBER

Grant 08-3066

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Dr. Lukas Burget

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Brno University of Technology
Bozetechova 2
Brno 61266
Czech Republic

8. PERFORMING ORGANIZATION REPORT NUMBER

Grant 08-3066

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
Unit 4515 BOX 14
APO AE 09421

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-AFOSR-UK-TR-2010-0016

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking Brno University of Technology as follows: TECHNICAL PROPOSAL/DESCRIPTION OF WORK:

The total duration of the project is divided into 2 phases
The first phase planned for the period May 2008 to Oct 2008
The second phase planned for Nov 2008 to April 2008.
It has the following 3 work-packages (WP).

WP1 Acquisition and selection of the data (Phase 1)

This project counts on Voice of America (VOA) data collection performed by LDC in the several past years. The VOA data will need to be completed with the available meta-information, especially about the language(s) contained. The following step will consist of cleaning the data and selecting relevant speech information, as we are aware of the automatically acquired data being quite dirty for the purposes of LRE:
1. automatic segmentation into speech, music and noise segments, while only speech will be retained. The speech/music segmentation was the topic of a diploma thesis finished at our department [Hovorka2006] and is available for use in this project.
2. voice activity detection (VAD) that will be performed by our phoneme recognizer [Schwarz2006] with all phoneme classes linked to 'speech' class. This setup was successfully used in a wide range of applications such as speaker recognition, language recognition, speech transcription and spoken term detection and evaluated in several NIST evaluations.
3. detecting telephone conversations in the data. In this project, we will mainly investigate the data that is as closed as possible to the target domain: conversational telephone speech (CTS). Therefore, we will concentrate on the segments with detected telephone speech (people calling in the broadcast) as we believe these should correspond the best to CTS. Initial work on Thai done for NIST LRE 2007 has shown a yield of 8 hours of telephone conversations from approximately 400 hours of VOA data downloaded from the Internet archive of VOA. However, we will tag and store also the general speech data (news, ads, talk-shows, etc.), as we believe this should be also representative of the target language – this data can then be used in the second part of this project and in eventual follow-up(s).
In all three steps, we count on cleaning the data quite aggressively, as the amount of available broadcast data is practically unlimited.

WP2 Training and testing in LRE systems (Phase 1)

The acquired data will be used to train our LRE systems. We count on running at least one acoustic and one phonotactic system out of the set

of 13 systems submitted by BUT to NIST LRE 2007 evaluations [BUT-NIST-LRE-2007]. The systems trained on automatically acquired data will be tested on standard and well known test data (NIST 2005) and their performance will be compared to the training on standard speech corpora (as is the current state). We will also simulate the “unseen language” scenario by not using any standard data for one or more selected languages and training these using automatically acquired data only. These tests will be performed only on telephone conversations extracted from the broadcasts.

The second step of this WP will be devoted to testing the current systems on the automatically acquired data. The results of tests will be compared to these obtained on standard telephone data. The goal is to see whether the performance improvements of LRE systems seen on such automatically acquired data generalize also to standard telephone data. If this is the case, there is a chance to make the data collection for future evaluations significantly cheaper and easier, as it will be possible to collect the necessary data from radios.

WP3 Channel compensation (Phase 2)

First results obtained on Thai in the NIST LRE 2007 evaluation have shown, that without proper channel adaptation, the system works, but the accuracy is significantly worse than when trained on standard spontaneous telephone data. We were extensively working on channel compensation for both speaker [Brummer2007,Burget2007] and language [BUT-NIST-LRE-2007] recognition. The outcomes of this work will be used to improve the accuracy of LRE system trained on automatically acquired data in the following ways:

1. the spectral properties of telephone speech occurring in the broadcast will be investigated and an a-priori compensation will be applied.
2. eigen-channel compensation techniques (in feature and model domain) will be used, simply by considering the radio-recorded data as an additional transmission channel to the channels already present in our system.
3. the differences in spontaneity, lack of hesitations and other psycho-linguistic factors related to different scenario (calling in a broadcast vs. spontaneous communication) will be addressed by phonotactic model compensation techniques, such as latent factor analysis, already studied in our work for NIST LRE 2007.

We also propose to build on our work in discriminative training – this approach should inherently remove everything but the information on the target class from the decision process. A thorough comparison in both above mentioned scenarios (e.g. augmenting the data for a known language vs. modeling “unknown” language) will be performed.

FACILITIES/EQUIPMENT:

the current computing infrastructure of Faculty of information technology of BUT is going to be used for the project. The broadcast data will be supplied by Linguistic Data Consortium http://www.ldc.upenn.edu

SCHEDULE OF REPORTS/DELIVERIES:

The project is proposed for 12 months, with two 6-month phases. The approximate schedule of the project is as follows:

Phase 1:
M1 agreement on the data, acquisition of the data from LDC.
M2 generating/checking the meta-information coming with the data.
M2-M3 developing, testing and refining procedures for selecting suitable data.
M3-M6 training and testing with the current acoustic system, basic adaptation.
M3-M6 training and testing with the current phonotactic system, basic adaptation.
M6 intermediate report to the sponsor.

Phase 2:
M7-M8 refinement and testing of acoustic channel compensation techniques, training, testing.
M8-M9 - refinement and testing of phonotactic model compensation techniques, training, testing.
M9-M12 tests with full (not only telephone) data.
M12 final report.

15. SUBJECT TERMS
EOARD, Linguistics, Language

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** TAMMY SAVOIE, Lt Col, USAF
a. REPORT UNCLAS	**b. ABSTRACT** UNCLAS	**c. THIS PAGE** UNCLAS	**SAR**	46	**19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)1895 616459

Improving the capacity of language recognition systems to handle rare languages using radio broadcast data

Final report of project supported by EOARD under Grant No. 083066

Lukáš Burget, Oldřich Plchot, Valiantsina Hubeika, Petr Schwarz, Pavel Matějka
and Jan “Honza” Černocký

Speech@FIT, Brno University of Technology, Czech Republic,
{burget,iplchot,ihubeika,schwarzp,matejkap,cernocky}@fit.vutbr.cz

December 2010

Contents

List of Figures

List of Tables

Summary

Current LID systems have difficulties in dealing with languages with insufficient or small amount of training data available. This issue concerns not only exotic languages with small number of native speakers, but also languages like Thai with 65 million native speakers.

We aim to develop techniques, that will allow us to automatically obtain training data for these troubled languages and use them in Language Recognition systems. As the present LRE systems are trained and evaluated on Continuous Telephone Speech (CTS), the task will be to obtain speech samples, that went through the telephone channel. This task leads us to developing an automatic system, which obtains recordings from public broadcasts and automatically detects telephone calls that are consequently used for training. The system was implemented and used for building the data sets which were used for subsequent experiments.

In order to use the data obtained from broadcasts, we have to cope with several issues related to this data. The first problem is channel compensation, as the data comes not only through telephone channel, but also through wide-band broadcast. The second problem is that the telephone calls into broadcasts are usually less spontaneous than data commonly used for current systems.

We have conducted several experiments using both CTS and broadcast data to uncover possible problems, which can arise when using this type of data in training or evaluating current LRE systems. The results of these experiments show that if the broadcast data only are used for training and standard telephone data for testing, the performance of such system is worse, than the performance of standard LRE systems trained and tested on CTS.

The experiments also show, that if the broadcast data are used both for training and testing the system, the results are very good. This can indicate, that the information about channel is very strong in these broadcast data and that the systems are learning this information and it heavily affects the final recognition.

Cooperation with Linguistic Data Consortium on creating a broadcast database was part of this work. We used the developed systems to provide pre-labeling of broadcast data, see Appendix C.

Acknowledgments

This work was supported by US Air Force European Office of Aerospace Research & Development (EOARD) under Grant No. 083066.

Chapter 1

Introduction

We introduce a process of automatic acquisition of speech data from the various media sources for the language identification task. The last editions of NIST Language Recognition (LRE) evaluations have shown that both acoustic and phonotactic approaches have reached a certain maturity level in both modeling of target languages and dealing with the influences of different channels. However, we are still facing the common problem: the lack of training data. There is no good or large enough database of training data for many languages including even languages like Thai, which is spoken by 65 million speakers. Also, there is an increasing demand to recognize languages from smaller and less populous regions (many of them relevant for security of defense domain). For some of these languages, no standard speech resources exist.

This work aims at solving this problem using the data acquired from public sources, such as satellite and Internet TVs and radios, which contain conversational speech or telephone calls. This approach can provide us with large amount of data that we will use to conduct experiments, which will help to answer the question whether these data can replace or augment standard conversational telephone speech (CTS) data. The results will also show that if we had no standard CTS training data, these data obtained from broadcasts can be used to process the languages that we were unable to recognize due to absence of the training data.

First, the obtained data has to be preprocessed in order to acquire clean speech segments or individual phone calls. The task is to examine the obtained telephone calls by training and evaluating the systems on languages for which we have both CTS and broadcast data. The results of the experiments will show, how the systems perform, when the CTS or broadcast data are used for training or testing.

The main challenge is channel compensation, as the obtained data are acoustically very different from the conversational telephone speech (CTS) commonly used in LRE. Broadcast data contain a great deal of unspontaneous speech as well. Another task is to explore how unspontaneous speech affects current LRE systems (which are supposed to be trained on spontaneous data). The notion of channel compensation will therefore have to be extended to cope with these factors.

In the Phase 1 of the project, we have done experiments on Dari, English, French, Hindi, Korean, Mandarin, Spanish and Vietnamese languages, these languages are the intersection of languages we obtained from broadcast sources and the languages present in standard databases available.

After the initial experiments in the Phase 1 of the project, we have concentrated our work on advanced techniques such as Joint Factor Analysis (JFA) and i-vector based systems, which both can very well compensate for the channel variabilities. Also, we participated in the NIST 2009 LRE evaluation and consecutive workshop, where the general discussion addressed new problems related to the nature of broadcast data.

Many laboratories confirmed that a small speaker diversity (repeating speakers for training the system) can significantly decrease the performance of the LRE system. We have investigated this issue using our speaker verification system.

Another difference of the broadcast data to the standard conversational telephone speech (CTS) databases are the issues of spontaneity, hesitations and other psycho-linguistic factors. This is caused by presence of many professional speakers or speakers calling in a broadcast with a premeditated speech. We are investigating this character of the data by experimenting with inter-session variability compensation (PIVCO) techniques.

Chapter 2

Methods, Assumptions and Procedures

2.1 Data Acquisition Principles

There is unlimited source of speech data available from the broadcast media. We can acquire data from several sources, each of which has different channel parameters, quality and number of available languages. The list of available sources in a standard industrialized country (such as the Czech Republic) is shown in Table 2.1 [1].

All of the listed sources except Internet radios are geographically dependent regarding location. The quality of different Internet sources varies a lot and it is important to carefully choose them. We have used an archive[1] of Voice of America Internet radio to obtain data for all languages.

This particular data of VoA were obtained in MP3 format, bit-rate is 24 Kbit/s, sampling rate 22,050 Hz, 16 bit encoding, mono. Original media data include a great portion of music and speech with the music in background. We have to deal with this problem and select only clean speech segments. Also, we should deal with the problem of a low speaker variability in the obtained data, for instance as it is common in news programs, which are moderated by the same speaker. So far, we have not investigated into this problem and used only telephone calls in broadcasts, where speaker variability should be sufficient.

2.1.1 Detecting Phone Calls

Our phone call detector is based on the fact that a telephone channel acts like a band-pass filter, which passes energy between approximately 400 Hz and 3.4 KHz. On the other hand, regular wide-band speech contains significant energy up to around 5 KHz. Common media sources like satellite radio or Internet radios are usually sampled at 22 kHz so they support this bandwidth, which means that if we place a phone call into the regular radio transmission, we will see a significant change in the spectrum (Figure 2.1).

For the detection, we first re-sample the signal to commonly used 16 kHz. The signal is divided into frames of 512 samples with no overlap and Fourier spectrum is computed for each frame. To detect boundary between wide-band and telephone speech, we concentrate on the frequency range between 2350 and 4600 Hz. The power spectral density (PSD) in this range was used (see Figure 2.2). At first, the PSD was normalized to zero mean and unit variance. Then values in the first half (from 2350 to 3475 Hz) and values in the second half (from 3475 to 4600 Hz) of the PSD were summed. A ratio between these two sums was compared with a threshold and the decision was made. If the ratio is higher than selected threshold, there is more energy in lower frequencies and we considered the segment a telephone call speech. For the block diagram of this process, see Figure 2.3.

[1] FTP server 8475.ftp.storage.akadns.net directory /mp3/voa

Table 2.1: Overview of different channels. DVB stands for Digital Video Broadcasting - Terrestrial, Cable and Satellite. By parallel recording we mean the possibility of acquiring more broadcasts simultaneously using one recording device (i.e. one DVB-S receiver).

	Inet. radio	**DVB-T**	**DVB-C**	**DVB-S**	**Analog**
Languages	approx. 100	1 - 3	approx. 5	20 - 30	3 - 5
Quality	variable	good	good	good	bad
Parallel recording	yes	yes	yes	yes	no

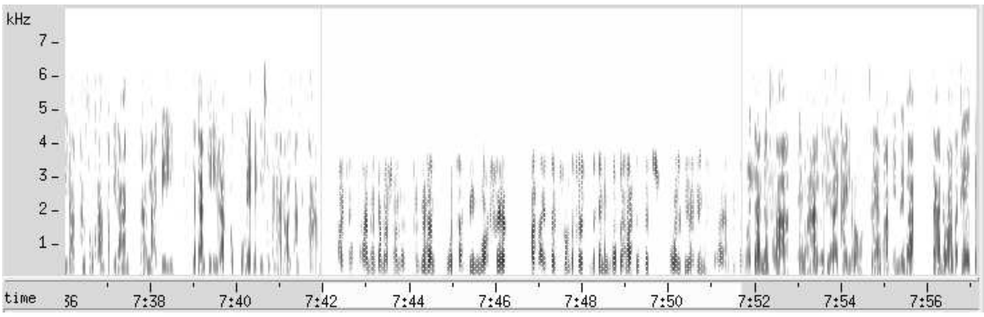

Figure 2.1: Phone Call in a Radio Broadcast.

2.1.2 Detecting Wide-band Speech Segments

Recordings obtained from media broadcasts contain great deal of music, speech with music in the background or other non-speech sounds. The task is to detect clean speech segments which can be used in language recognition or possibly in the other applications.

The detection is done by estimating frame by frame likelihoods, of classes *speech* and *other* (non-speech). GMM models were used to estimate these likelihoods. These models contain 1024 Gaussians and were trained on 12.7 hours of speech and 18.7 hours of non-speech wide-band data. MFCC coefficients with deltas and double deltas were used as features for training. These data (containing several languages) were obtained from Linguistic Data Consortium and were manually annotated for these two classes.

Once we obtain frame by frame log-likelihoods for each class, we filter them using simple median filter[2] and subtract these two sets of values. The resulting log-likelihood ratios are averaged over 100 frames and compared to empirically set thresholds. Depending on the threshold, we decide whether we are in the speech segment or non-speech segment or whether we are not sure (segments to be checked by human annotator). For the block diagram of this process see Figure 2.4.

2.2 Dealing with repeating speakers

Having the audio segments with repeating speakers in the training and development data sets, (the later being also used for training of the calibration and fusion parameters), causes over-training of the system. Especially for languages with little amount of data, this can cause large decrease in performance when testing on an evaluation data set, which does not include speakers who were seen in the training and calibration phases. This problem is especially serious for the acoustic systems, where we saw a huge drop in performance when comparing results on our development set with results on the evaluation set.

[2] Window size of this median filter is 5.

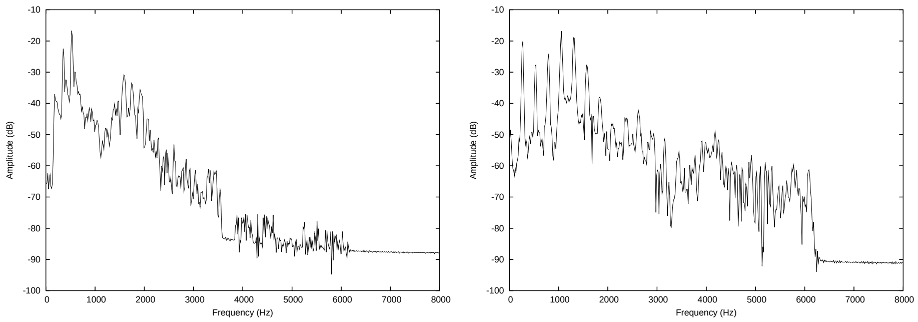

Figure 2.2: Power Spectral Density of telephone call in the broadcast (left figure) and wide-band speech (right figure).

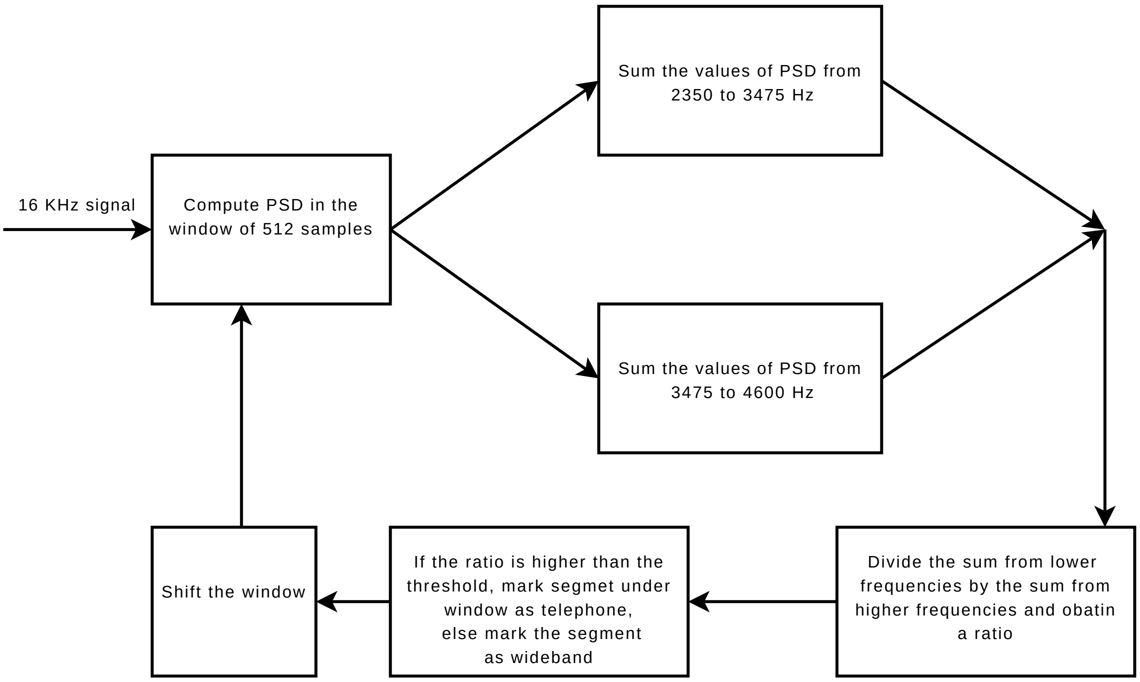
16 KHz signal
Compute PSD in the window of 512 samples
Sum the values of PSD from 2350 to 3475 Hz
Sum the values of PSD from 3475 to 4600 Hz
Divide the sum from lower frequencies by the sum from higher frequencies and obatin a ratio
If the ratio is higher than the threshold, mark segmet under window as telephone, else mark the segment as wideband
Shift the window

Figure 2.3: Block diagram of detecting telephone calls in the wide-band signal.

We decided to address this problem by training a speaker ID system for each training utterance and scoring all development utterances from the corresponding language.

2.3 Improved channel variability compensation

We followed a novel design for acoustic feature-based language recognizers [2]. Our design is inspired by recent advances in text-independent speaker recognition, where intra-class variability is modeled by factor analysis in Gaussian mixture model (GMM) space. We use approximations to GMM likelihoods, which allow variable-length data sequences to be represented as statistics of fixed size [3, 4, 5].

We use these statistics for all further computation, in both training and test. The advantage of this approach is an efficient implementation of speaker-recognition-style channel compensation. Specifically, we use a factor-analysis model for the kth component mean of the GMM for segment s:

$$\mathbf{m}_{sk} = \mathbf{t}_{l(s)k} + \mathbf{U}_k \mathbf{x}_s, \tag{2.1}$$

where $l(s)$ denotes the language of segment s; $\mathbf{t}_{lk}$ are language location vectors; $\mathbf{x}_s$ is a vector of C

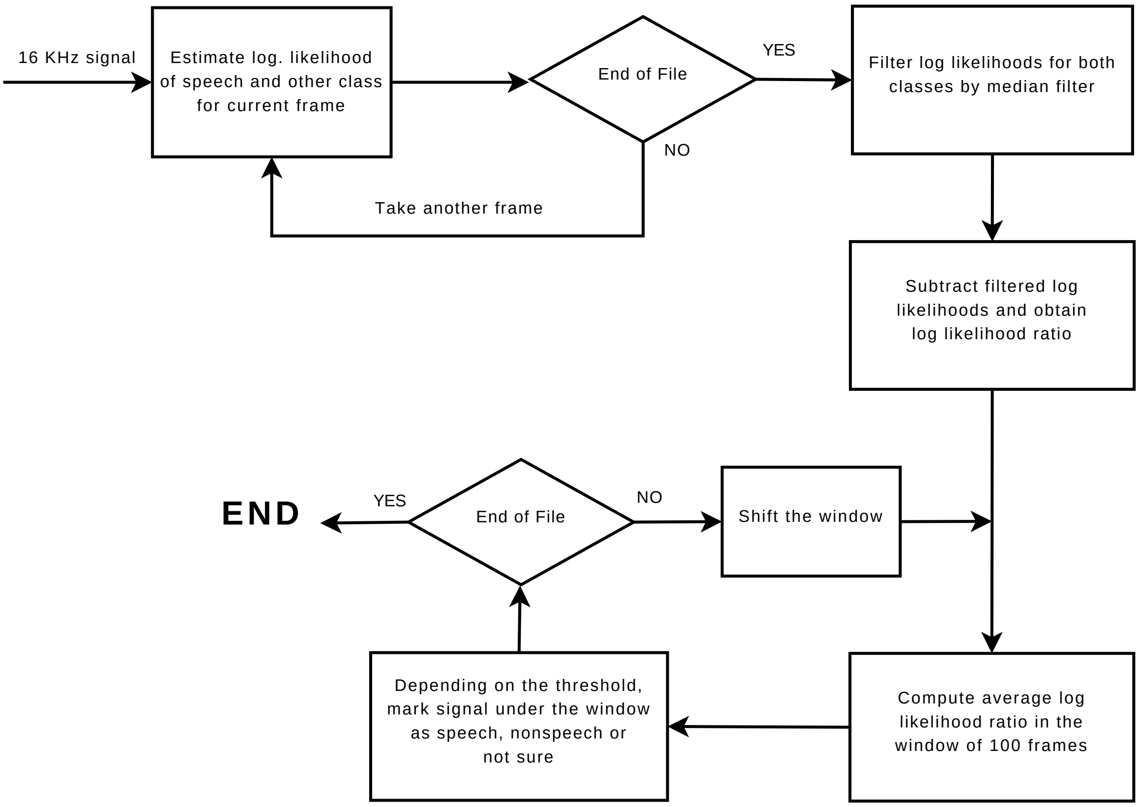
16 KHz signal
Estimate log. likelihood of speech and other class for current frame
End of File
YES
NO
Take another frame
Filter log likelihoods for both classes by median filter
Subtract filtered log likelihoods and obtain log likelihood ratio
END
YES
End of File
NO
Shift the window
Depending on the threshold, mark signal under the window as speech, nonspeech or not sure
Compute average log likelihood ratio in the window of 100 frames

Figure 2.4: Block diagram of detecting speech and non-speech segments in the wide-band signal.

segment-dependent *'channel factors'*; and $\mathbf{U}_k$ is a 56-by-C *factor loading matrix.* The channel factors are assumed to be drawn independently from the standard normal distribution. As in the case of the first-order statistics, we stack component-dependent vectors into super-vectors $\mathbf{m}_s$ and $\mathbf{t}_l$ and we stack the component-dependent $\mathbf{U}_k$ matrices into a single matrix $\mathbf{U}$, so that (2.1) can be expressed more compactly as:

$$\mathbf{m}_s = \mathbf{t}_{l(s)} + \mathbf{U}\mathbf{x}_s. \tag{2.2}$$

We refer to $\mathbf{U}$ as the *channel matrix.*

Following [3], we estimate the channel matrix with maximum likelihood, by using the EM-algorithm. We tested different sizes for $\mathbf{U}$ and found $C = 50$ to be a good choice. Then, we use all speech segments for all of the languages that we have available in our development set. Next, we apply channel compensation: Given the channel matrix $\mathbf{U}$ and the statistics $\mathbf{f}_{sk}, n_{sk}$ for a speech segment s and Gaussian component k, we perform language-independent maximum-a-posteriori (MAP) point-estimate of the channel factors $\mathbf{x}_s$ relative to the universal background model (UBM) [6, 7]. This estimate is computed as:

$$\hat{\mathbf{x}}_s = \left(\mathbf{I} + \sum_k n_{sk}\mathbf{U}_k'\mathbf{U}_k\right)^{-1} \mathbf{U}'\mathbf{f}_s. \tag{2.3}$$

Next, the effect of the channel factors can be approximately removed from the first-order statistics thus:

$$\tilde{\mathbf{f}}_{sk} = \mathbf{f}_{sk} - n_{sk}\mathbf{U}_k\hat{\mathbf{x}}_s. \tag{2.4}$$

We refer to $\tilde{\mathbf{f}}_{sk}$ as the *compensated first-order statistic.* In our experiments, we try both uncompensated and compensated statistics. We find the compensation to dramatically improve the accuracy.

2.4 Phonotactic intersession variation compensation

JFA has become state-of-the-art technique in speaker recognition and it has been also successfully applied to language recognition. The principle of this technique lies in probabilistic approach to modeling various types of target model parameter variability. These parameters are usually means of the GMM, but they can be also N-gram probabilities. The idea of our approach is to adapt this technique to *multinomial models.*

Multinomial distribution is a generalization of binomial distribution, specifying trials to result in one of some fixed finite number C of possible outcomes, with probabilities $\theta = (p_1, ..., p_C)$. All probabilities follow:

$$\forall i \quad p_i \in \langle 0, 1 \rangle. \tag{2.5}$$

$$\sum_{i=1}^{C} p_i = 1. \tag{2.6}$$

Formally, all parameters p_i lie on $C-1$ simplex, see Figures in Appendix B.

In the N-gram language model, words with the same history of $N > 1$ follow multinomial distribution. For example:

$$\Gamma = \{\text{swimming}, \text{party}, \text{pool}\}$$

$$\begin{aligned} &p(\text{pool}|\text{swimming}) \\ &+p(\text{party}|\text{swimming}) \\ &+p(\text{swimming}|\text{swimming}) \\ &= 1. \end{aligned}$$

Let the succession of N events be a matrix $\mathbf{\Gamma}$ of size $C \times N$, where columns are zeros except for the event index, which is 1. If we sum the columns of $\mathbf{\Gamma}$ (N-gram counts), we can write log-likelihood of data as

$$\log p(\boldsymbol{\gamma}|\theta) = \sum_{i=1}^{C} \gamma_i \log p_i. \tag{2.7}$$

In JFA, the aim was to find a parameter subspace (parameters were means of GMM), in which we could effectively adapt our model. Here, in multinomial model, the only parameters we can modify, are probabilities p_i. The restrictions $p_i > 0$, $\sum p_i = 1$ do not allow us to directly model these parameters as a linear combination of some kind.

What we can do, is to linearly combine parameters in the log domain:

$$q_i(\hat{\theta}) = m_i + \mathbf{u}_i \mathbf{x}, \tag{2.8}$$

where m_i can be set to $\log p_i$, $\mathbf{u}_i$ is the ith row of a factor loading matrix $\mathbf{U}$ and $\mathbf{x}$ is a vector of factors. The second condition $\sum p_i = 1$ can be enforced by normalizing in the linear domain:

$$\omega_i(\hat{\theta}) = \frac{q_i}{\sum_{\hat{i}=1}^{C} q_{\hat{i}}}, \tag{2.9}$$

and ω_i will now be correct parameters of multinomial distribution.

We can now rewrite (2.7) as

$$\log p(\boldsymbol{\gamma}|\hat{\theta}) = \sum_{i=1}^{C} \gamma_i, \log \omega_i(\hat{\theta}) \tag{2.10}$$

where $\boldsymbol{\omega}$ is a valid probability distribution given by

$$\omega_i(\hat{\theta}) = \frac{e^{m_i + \mathbf{u}_i \mathbf{x}}}{\sum_{\hat{i}=1}^{C} e^{m_{\hat{i}} + \mathbf{u}_{\hat{i}} \mathbf{x}}} \tag{2.11}$$

and $\hat{\theta}$ represents all parameters in the exponent. Channel subspace is then represented by a curve laying on a simplex, see Figure B.2.

When training the system, we use N-gram probabilities to define the model parameter space, then we search for their subspace, which best describes the inter-session variability. In the test phase, we let the model adapt to the test utterance in this subspace, see [8] for details.

2.4.1 Using binary trees to obtain N-gram probabilities

It was shown [9, 10], that clustering the N-gram history by using binary decision trees (BT) improves the performance. Growing the tree is based on finding questions about the history, following the maximum entropy reduction (or likelihood increase) criterion. Each of these questions clusters the data into two subsets. The conditional probabilities are then stored in the leaves and are estimated from the clustered data. Two approaches to BT estimation are proposed—building the whole tree for each class in one case, and adapting from a UBM in the other case. We have adopted the latter framework and used it in conjunction with other techniques, see [8] for details.

2.5 Channel compensation in a low dimensional i-vector space

Recent results in the NIST SRE (Speaker Recognition) evaluations demonstrated, that using a single low-dimensional space for modeling both speaker and channel variability can improve and simplify state-of-the-art speaker verification systems. In this approach inspired by the Joint Factor Analysis framework introduced in [3, 11], we model the *total variability* or *i-vector* space using a simple factor analysis [12]. By applying this technique, we are able to reduce the large-dimensional input data to a low-dimensional feature vector while retaining most of the relevant information. We have successfully applied this technique to the language identification task, where, instead of variability between speakers, we model the variability between languages.

2.5.1 Theoretical Background for Extracting i-vectors

Let us first state the motivation for the i-vectors. The main idea is that the language- and channel-dependent GMM super-vector $\mathbf{s}$ can be modeled as:

$$\mathbf{l} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{2.12}$$

where $\mathbf{m}$ is the UBM GMM mean super-vector, $\mathbf{T}$ is a low-rank matrix representing M bases spanning subspace with important variability in the mean super-vector space, and $\mathbf{w}$ is a standard normal distributed vector of size M.

For each observation $\mathcal{X}$, the aim is to estimate the parameters of the posterior probability of $\mathbf{w}$:

$$\mathrm{p}(\mathbf{w}|\mathcal{X}) = \mathcal{N}(\mathbf{w}; \mathbf{w}_{\mathcal{X}}, \mathbf{L}_{\mathcal{X}}^{-1}) \tag{2.13}$$

The i-vector is the MAP point estimate of the variable $\mathbf{w}$, i.e. the mean $\mathbf{w}_{\mathcal{X}}$ of the posterior distribution $\mathrm{p}(\mathbf{w}|\mathcal{X})$. It maps most of the relevant information from a variable-length observation $\mathcal{X}$ to a fixed-(small-) dimensional vector. $\mathbf{T}$ is referred to as the i-vector extractor.

Data

The input data for the observation $\mathcal{X}$ is given as a set of *zero-* and *first-order statistics* — $\mathbf{n}_{\mathcal{X}}$ and $\mathbf{f}_{\mathcal{X}}$. These are extracted from F-dimensional features using a GMM UBM with C mixture components, defined by a mean super-vector $\mathbf{m}$, component covariance matrices $\mathbf{\Sigma}^{(c)}$, and a vector of mixture weights $\boldsymbol{\omega}$. For each Gaussian component c, the statistics are given respectively as:

$$N_{\mathcal{X}}^{(c)} = \sum_t \gamma_t^{(c)} \tag{2.14}$$

$$\mathbf{f}_{\mathcal{X}}^{(c)} = \sum_t \gamma_t^{(c)} \mathbf{o}_t \tag{2.15}$$

where $\mathbf{o}_t$ is the feature vector in time t, and $\gamma_t^{(c)}$ is its occupation probability. The complete zero- and first-order statistics super-vectors are $\mathbf{f}_{\mathcal{X}} = \left(\mathbf{f}_{\mathcal{X}}^{(1)'}, \dots, \mathbf{f}_{\mathcal{X}}^{(C)'}\right)'$, and $\mathbf{n}_{\mathcal{X}} = \left(N_{\mathcal{X}}^{(1)}, \dots, N_{\mathcal{X}}^{(C)}\right)'$.

For convenience, we *center* the first order statistics around the UBM means, which allows us to treat the UBM means effectively as a vector of zeros:

$$\begin{aligned} \mathbf{f}_{\mathcal{X}}^{(c)} &\leftarrow \mathbf{f}_{\mathcal{X}}^{(c)} - N_{\mathcal{X}}^{(c)} \mathbf{m}^{(c)} \\ \mathbf{m}^{(c)} &\leftarrow \mathbf{0} \end{aligned}$$

Similarly, we "normalize" the first-order statistics and the matrix $\mathbf{T}$ by the UBM covariances, which again allows us to treat the UBM covariances as an identity matrix[3]:

$$\begin{aligned} \mathbf{f}_{\mathcal{X}}^{(c)} &\leftarrow \mathbf{\Sigma}^{(c)-\frac{1}{2}} \mathbf{f}_{\mathcal{X}}^{(c)} \\ \mathbf{T}^{(c)} &\leftarrow \mathbf{\Sigma}^{(c)-\frac{1}{2}} \mathbf{T}^{(c)} \\ \mathbf{\Sigma}^{(c)} &\leftarrow \mathbf{I} \end{aligned}$$

where $\mathbf{\Sigma}^{(c)-\frac{1}{2}}$ is a Cholesky decomposition of an inverse of $\mathbf{\Sigma}^{(c)}$, and $\mathbf{T}^{(c)}$ is a $F \times M$ sub-matrix of $\mathbf{T}$ corresponding to the c mixture component such that $\mathbf{T} = \left(\mathbf{T}^{(1)'}, \dots, \mathbf{T}^{(C)'}\right)'$.

Parameter Estimation

As described in [11] and with the data transforms from previous section, for an observation $\mathcal{X}$, the corresponding i-vector is computed as a point estimate:

$$\mathbf{w}_{\mathcal{X}} = \mathbf{L}_{\mathcal{X}}^{-1} \mathbf{T}' \mathbf{f}_{\mathcal{X}}, \tag{2.16}$$

where $\mathbf{L}$ is the precision matrix of the posterior distribution, computed as:

$$\mathbf{L}_{\mathcal{X}} = \mathbf{I} + \sum_{c=1}^{C} N_{\mathcal{X}}^{(c)} \mathbf{T}^{(c)'} \mathbf{T}^{(c)}. \tag{2.17}$$

The computational complexity of the whole estimation for one observation is $O(CFM + CM^2 + M^3)$. The first term represents the $\mathbf{T}'\mathbf{f}_{\mathcal{X}}$ multiplication. The second term represents the sum in (2.17) and includes the multiplication of $\mathbf{L}_{\mathcal{X}}^{-1}$ with a vector. The third term represents the matrix inversion.

The memory complexity of the estimation is $O(CFM + CM^2)$. The first term represents the storage of all the input variables in (2.16), and the second term represents the pre-computed matrices in the sum of (2.17).

Note that the computation complexity grows quadratically with M in the sum of (2.17), and linearly with C. This becomes the bottle-neck in the i-vector computation, resulting in high memory and CPU demands.

[3] Part of the factor estimation is a computation of $\mathbf{T}'\mathbf{\Sigma}^{-1}\mathbf{f}$, where the decomposed $\mathbf{\Sigma}^{-1}$ can be projected to the neighboring terms, see [11] for detailed formulae.

Model Training

Model hyper-parameters $\mathbf{T}$ are estimated using the same EM algorithm as in case of JFA [11]. Note that our algorithm makes use of an additional *minimum divergence* update step [13, 14], which yields a quicker convergence, but is not described here.

In the E-step, the following accumulators are collected using all training observations i:

$$\mathbf{C} = \sum_i \mathbf{f}_i \mathbf{w}_i' \tag{2.18}$$

$$\mathbf{A}^{(c)} = \sum_i N_i^{(c)} \left(\mathbf{L}_i^{-1} + \mathbf{w}_i \mathbf{w}_i' \right) \tag{2.19}$$

where $\mathbf{w}_i$ and $\mathbf{L}_i$ are the estimates from (2.16) and (2.17) for observation i. The M-step update is given as follows:

$$\mathbf{T}^{(c)} = \mathbf{C}\mathbf{A}^{(c)^{-1}} \tag{2.20}$$

2.6 Training and Test Sets used in Phase 1

In order to compare, how our LRE systems perform when using broadcast and standard CTS data, we created a data set from broadcast data. We selected eight languages[4] from the Voice of America ftp archive. We have chosen these particular languages, because we have the data for these languages present in CallFriend, NIST LRE 2003 and NIST LRE2007 databases. In order to create reasonably robust experiment, we have chosen these languages even if we expected problems with French and Dari language: the French language in the Voice of America archive is recorded in the Africa region and therefore the obtained samples can substantially differ from the utterances spoken by native French speakers in our CTS databases. The *Dari* language was chosen, because this language is very close to the *Farsi* language which is present in CallFriend, NIST LRE 2003 and NIST LRE 2007 databases. We decided to relabel **Farsi to Dari** in those databases for the purpose of the experiments.

Additionally, we expect, that the people calling into the Voice of America broadcasts *speak the same language* as the language label denoting particular recording of broadcast. We did not have resources to manually check all data, so errors can occur in labeling of the training and test data. We have to keep in mind all of these compromises we have made when analyzing the results of the experiments.

2.6.1 Telephone Call Segments

We decided to select only telephone calls which are present in the Voice of America broadcasts, because we believe these data will be affected by passing through the telephone channel and will better match our CTS data. First, our phone call detector was used to *detect phone call segments* in the wide-band data. The telephone call into broadcast can be interrupted by a moderator and we want to reconstruct the call from the segments of the calling person. A post-processing of this detection was made in order to obtain these reconstructed segments.

For the purpose of the post-processing of label file created by phone detector, an algorithm which marks particular phone segments as `phonecall1, phonecall2` ... was designed. This algorithm marks individual phone call segments in order to join them into longer segments. The algorithm accepts segments which are longer than *10 seconds*, because our phone call detector makes a lot of short segments, which are more likely to contain some wide-band portion. Phone call segments are assigned the same label until there is a maximum *120 seconds* of wide-band segment between them.

[4] Dari, English, French, Hindi, Korean, Mandarin, Spanish and Vietnamese

Table 2.2: Training data in hours after segmentation for each language.

Language	CallFriend	Broadcast
Dari/Farsi	21.2	6
English	39.8	6
French	21.5	6
Hindi	19.6	6
Korean	18.4	6
Mandarin	41.7	6
Spanish	43.8	6
Vietnamese	20.6	6

When the wide-band segment between phone calls is longer than 240 seconds, the next phone segments will be assigned new label (e.g. `phonecall2`).

When the label file created by the telephone detector is processed by the algorithm explained above, we cut and join the segments with the same label. BUT phone recognizer [15], [16] was used to determine the pause in the speech at the borders of each segment and these time stamps were used to cut the segments out of the original recordings. Then the cut segments with the same label were concatenated into one file to obtain the reconstructed telephone call.

Using this approach, we obtain significantly smaller number of telephone segments than we would get taking directly the output of the telephone detector. The benefit is that the segments contain less wide-band caused by errors in detecting the phone calls and the speaker variability is increased, because we have less segments with the same speaker. On the other hand, it is possible, that the final segments contain more different speakers.

2.6.2 Broadcast Data Sets

Using the procedure explained above, we created *broadcast test set*, selecting 150 segments for each language. Each selected segment was cut out from the detected telephone call in such way, that it contained *30 seconds* of speech. Our phone recognizer was used to determine the length of speech.

Broadcast training set was created by taking the merged phone call segments[5] until we reached the limit of six hours of speech per language.

2.6.3 CTS Data Sets

CTS test sets were created by taking subsets of NIST LRE 2003 [17] and 2007 [18] evaluation data. Only *30 second* segments were used. Training set was created by taking subset of languages from CallFriend database. All data sets are listed in tables 2.2 and 2.3.

2.7 Training and Test Sets used in Phase 2

The motivation to use new data sets for experiments with more advanced systems in Phase 2 of the project was to obtain results on a more complex, more challenging and publicly defined task. The ideal solution was to report results on NIST LRE 2007 and 2009 task. As we took part in both 2007 and 2009 NIST LRE evaluations, we have used the same training and test sets as in our original NIST LRE submissions.

Table 2.4 lists the corpora (distributed by LDC and ELRA) used to train our systems.

[5] Described in section 2.6.1

Table 2.3: Number of 30 second test segments for each language.

Language	NIST 2003	NIST 2007	Broadcast
Dari/Farsi	80	88	150
English	240	266	150
French	80	80	150
Hindi	80	268	150
Korean	80	108	150
Mandarin	80	496	150
Spanish	80	256	150
Vietnamese	80	168	150

Table 2.4: Training databases for LRE2007 and LRE2009 systems

CF	CallFriend
CH	CallHome
F	Fisher English Part 1.and 2.
F	Fisher Levantine Arabic
F	HKUST Mandarin
SRE	Mixer (data from NIST SRE 2004, 2005, 2006, 2008)
LDC07	development data for NIST LRE 2007
OGI	OGI-multilingual
OGI22	OGI 22 languages
FAE	Foreign Accented English
SpDat	SpeechDat-East[6]
SB	SwitchBoard
VOA	Voice of America radio broadcast

2.7.1 Data sets for the experiments on NIST LRE 2007

Table 2.5 lists the training data in hours for each language and source database. Our development and test set were based on segments from previous NIST LRE evaluations plus additional segments extracted from longer files in the training databases, which were not contained in the training set.

2.7.2 Data sets for the experiments on NIST LRE 2009

Table 2.6 lists a detailed breakdown of the amounts of training data per language and source.

Our data was separated into two independent subsets, which we denoted TRAIN and DEV. The TRAIN subset had 54 languages (including the 23 target languages) and had about 80 000 segments in total. The DEV subset had 57 languages (including the 23 targets) and a total of about 60 000 segments. The DEV subset was split into balanced subsets having nominal durations of 3s, 10s and 30s. The DEV set was based on segments from previous evaluations plus additional segments extracted from longer files from CTS and VOA databases (which were not contained in the TRAIN set).

Table 2.5: Training data in hours for each language and source.

	sum	CF	CH	F	SRE	LDC07	OGI	OGI22	Other
Arabic	212	19.5	10.4	175	5.93	1.45		0.33	
Bengali	4.27				2.86	1.42			
Chinese	93.2	41.7	1.64	17.2	44.9	4.2	0.87	0.85	
English	264	39.8	4.68	162	34.9		6.77	0.52	15.6 (FAE)
Hindustani	23.5	19.6			0.64	1.32	1.53	0.42	
Spanish	54.3	43.8	6.71		2.63		1.18	0.38	
Farsi	22.7	21.2			0.03		1.00	0.42	
German	28.2	21.6	5.10				1.12	0.38	
Japanese	23.9	19.1	3.47				0.87	0.35	
Korean	19.7	18.4			0.09		0.72	0.5	
Russian	15.1				3.38	1.33		0.43	10.0 (SpDat)
Tamil	19.6	18.4					0.96	0.26	
Thai	1.45				0.15	1.23			
Vietnamese	21.6	20.6					0.79	0.27	
Other	62.5	20.7					1.10	3.29	37.4 (SpDat)

Table 2.6: Training data for NIST LRE 2009 in hours for each language and source.

Language	CTS		VOA	
	#files	#hours	#files	#hours
alba	0	0	104	3.4
amha	0	0	1724	77.7
arab	4085	201.8	0	0
azer	0	0	510	29.3
bang	213	5.2	3871	83.4
bosn	0	0	268	7.0
burm	0	0	3365	81.6
cant	482	6.9	34	2.1
creo	0	0	425	14.8
croa	0	0	150	5.3
czec	241	0.3	0	0
dari	0	0	2410	78.8
engi	714	2.2	0	0
engl	10560	290.9	3963	132.5
fars	656	22.6	0	0
fren	403	21.8	3679	88.7
geor	0	0	100	4.7
germ	685	23.1	0	0
gree	0	0	851	16.6
haus	0	0	2599	74.4
hind	755	26.0	358	15.7
hung	287	0.4	0	0
chin	1226	29.9	0	0
indo	267	0.4	226	3.0
ital	294	1.3	0	0
japa	718	23.1	0	0
khme	0	0	1297	53.0
knkr	0	0	1307	66.7
kore	691	21.3	342	16.3
mace	0	0	344	15.1
mand	1321	64.8	1049	35.7
ndeb	0	0	945	64.4
orom	0	0	399	15.1
pash	0	0	6317	102.3
pers	0	0	1673	70.6
poli	284	0.4	0	0
port	294	0.5	1069	48.7
russ	643	8.4	3071	82.2
serb	0	0	175	2.9
shon	0	0	553	58.6
soma	0	0	1909	70.9
span	1001	47.5	1623	67.6
swah	194	0.3	1965	70.9
swed	290	0.5	0	0
taga	24	0.6	0	0
tami	623	19.6	0	0
thai	209	6.6	0	0
tibe	0	0	349	2.0
tigr	0	0	395	24.6
turk	0	0	262	9.8
ukra	0	0	105	3.0
urdu	24	1.4	1242	67.2
uzbe	0	0	241	3.5
viet	743	25.7	113	8.9
SUM	27927	853.7	51382	1696.8

Chapter 3

Results and Discussion

We performed experiments both with phonotactic and acoustic systems. With both systems, we tested several techniques to improve the performance to show in which direction the development of LRE systems using data obtained from broadcasts together with standard CTS data should continue. The results are evaluated using standard metrics: Detection Error Tradeoff (DET) curve, Decision Cost Function (DCF) and Equal Error Rate (EER) [18]. All experiments were done on *30 second segments*. We present results of phonotactic and acoustic systems derived from our systems submitted to NIST LRE 2007 evaluation [19, 8].

3.1 Phonotactic Systems

The first phonotactic system [19, 8] is based on string output of our Hungarian phoneme recognizer. The second phonotactic system [19, 8] is based on lattice output of our Hungarian phoneme recognizer. The phoneme recognizer is based on hybrid ANN/HMM approach, where artificial neural networks (ANN) are used to estimate posterior probabilities of phonemes from Mel filter bank log energies using the context of 310ms around the current frame [16]. Trigram language models were trained on CallFriend database for CTS phonotactic system and for broadcast phonotactic system, the language models were trained on broadcast training set. Linear back-end calibration [20] was applied on the obtained scores. Calibration of scores was done on the test set, which may lead to overoptimistic results, but according to our experience, the results for properly trained calibration will not differ much. Both CTS and broadcast systems were evaluated against all test sets.

3.1.1 Results of Phonotactic Systems

The results are listed in tables 3.1 and 3.2. Phonotactic system based on string output was outperformed by the phonotactic system with lattices in all cases.

Table 3.1: Phonotactic systems based on string output - pooled EER

		TEST		
TRAIN		NIST 2003	NIST 2007	Broadcast
	CTS	1.781	9.072	6.583
	Broadcast	11.949	18.593	1.416

Table 3.2: Phonotactic Systems Based on Lattice Output - Pooled EER

		TEST		
		NIST 2003	NIST 2007	Broadcast
TRAIN	CTS	0.900	6.995	5.232
	Broadcast	8.958	15.215	1.398

3.2 Acoustic Systems

Our acoustic systems are built on the experience with GMM modeling for speaker recognition [21] which follows conventional Universal Background Model-Gaussian Mixture Modeling (UBM-GMM) paradigm [22] and employs number of techniques that have previously proved to improve GMM system performance [7]. This system was chosen because it can easily compensate for the channel distortion.

Table 2.2 lists the corpora used to train our systems. CTS system was trained on CallFriend database and broadcast system was trained on our broadcast database.

Our systems use the popular shifted-delta-cepstra (SDC) [23] feature extraction, where 7 MFCC coefficients (including coefficient C0) are concatenated with SDC 7-1-3-7, which totals in 56 coefficients per frame. Vocal-tract length normalization (VTLN) [24] performs simple speaker adaptation. VTLN warping factors are estimated using single GMM (512 Gaussians), ML-trained on the whole CallFriend database (using all the languages). The model was trained in standard speaker adaptive training (SAT) fashion in four iterations of alternately re-estimating the model parameters and the warping factors for the training data. Each language model is obtained by traditional *relevance MAP* adaptation [25] of UBM using enrollment conversation. Only means are adapted.

In the verification phase, standard Top-N Expected Log Likelihood Ratio (ELLR) scoring [25] is used to obtain verification score, where N is set to 10. However, for each trial, both language model and UBM are adapted to channel of test conversation using simple eigenchannel adaptation [21] prior to computing the log likelihood ratio score.

Calibration of scores was done on the test set, which may lead to overoptimistic results, but according to to our experience, the results for properly trained calibration will not differ much. Both CTS and broadcast systems were evaluated against all test sets.

3.2.1 Results of Acoustic Systems

First, both systems were trained without channel compensation. Then, eigenchannel adaptation was applied. Two different matrices containing 50 eigenchannels were used. The first matrix was computed from broadcast training set. The second matrix was taken from our NIST LRE2007 system [19]. This matrix was trained on CTS databases.

We also experimented with training channel compensation using both CTS and data from broadcasts, hoping that the channel compensation will solve the mismatch between CTS and broadcasts. Especially we were hoping to improve the poor results when training on broadcasts and testing on CTS. However, so far we were not successful with such cross-condition channel compensation.

The results are listed in tables 3.3, 3.4 and 3.5.

Table 3.3: Acoustic systems without eigenchannel compensation - pooled EER

		TEST		
		NIST 2003	NIST 2007	Broadcast
TRAIN	CTS	3.407	8.807	8.261
	Broadcast	14.423	19.502	3.250

Table 3.4: Acoustic systems with eigenchannels trained on broadcast data - pooled EER

		TEST		
		NIST 2003	NIST 2007	Broadcast
TRAIN	CTS	1.145	5.644	8.250
	Broadcast	9.840	15.013	0.583

3.3 Discussion

The results of both acoustic and phonotactic systems were consistent. Phonotactic systems using lattices significantly outperform phonotactic systems based on string output in all test cases. See Appendix A for detailed results.

We expected that the acoustic systems outperform phonotactic systems, but only phonotactic system trained on CTS was outperformed by acoustic system trained on CTS with channel compensation trained on telephone data.

The results of acoustic systems prove that the individual samples are recorded over different channels, therefore an application of eigenchannel adaptation [26] is crucial to compensate the channel distortion. In language detection task, channel variability may comprehend not only variability in the telephone channel or type of microphone, but also session or speaker variability.

Channel compensation trained on CTS is generally better. Broadcast data probably do not reflect the variations of channels.

The results of acoustic systems trained on broadcast data can imply, that the wide-band channel added additional distortion to the obtained data, which affects the results obtained when testing against the CTS data. The decline in performance when testing against the CTS data can be also affected by different type of speech, that is usually present in the broadcasts. Speech in media broadcasts is usually less spontaneous. Speech in radio broadcasts in comparison with our CTS databases does not contain many hesitations, interruptions and is usually grammatically correct.

However, the performance of systems trained on broadcast data and tested on CTS data is worse than the performance of systems trained and tested on CTS, the results show the similar trend over individual languages. This trend when EER is approximately two times higher except for the Dari and French language [1], can be observed on NIST 2007 test set (see figures B.3 and B.4), which consists of more difficult data for recognition.

[1] We expected problems for these languages, see section 2.6.

Table 3.5: Acoustic systems with eigenchannels trained on CTS data - pooled EER

		TEST		
TRAIN		NIST 2003	NIST 2007	Broadcast
	CTS	0.420	4.296	3.083
	Broadcast	9.222	14.290	0.922

When evaluating the acoustic system trained on broadcast data, we obtain excellent performance on broadcast data, which can indicate, that the system learned also the different channels of individual radio stations. This hypothesis has to be kept in mind when using broadcast data both for training and testing. Channel compensation trained on broadcasts even emphasizes this possible problem.

3.4 Experiments addressing repeating speakers issue

To evaluate the effect of speaker filtering, we selected JFA system based on Region Dependent Linear Transforms, which is the most affected acoustic subsystem in our NIST 2009 submission [27]. A GMM-UBM based speaker ID system developed by BUT for NIST 2006 SRE evaluation was used [28][2].

Based on the histogram of scores (example for Ukrainian in Figure 3.1) showing clearly bi-modal structure of identical and different speakers, we chose a language-dependent threshold of speaker ID scores for omitting utterances from the development set. The amount of omitted data is in Table 3.7. This step brings a nice improvement as we can see in Table 3.6.

Table 3.6: Fixing the data.

Eval data, $[C_{avg}]$	30s	10s	3s
JFA-G2048-RDLT	3.56	6.36	16.14
+ speaker ID filtration	2.33	5.09	15.06

Table 3.7: Amount of omitted data by speaker ID filtering

language	acronym	omitted data
Bosnian	bosn	92.8 %
Croatian	croa	77.9 %
Portuguese	port	17.6 %
Russian	russ	30.1 %
Ukrainian	ukra	93.8 %

3.5 Experiments with Improved Channel Compensation

Our baseline system used no channel compensation. We used uncompensated segment statistics to make relevance-MAP estimates of the language locations and uncompensated statistics to score new

[2] This system is available through Phonexia `http://phonexia.com/download/demo-sid`.

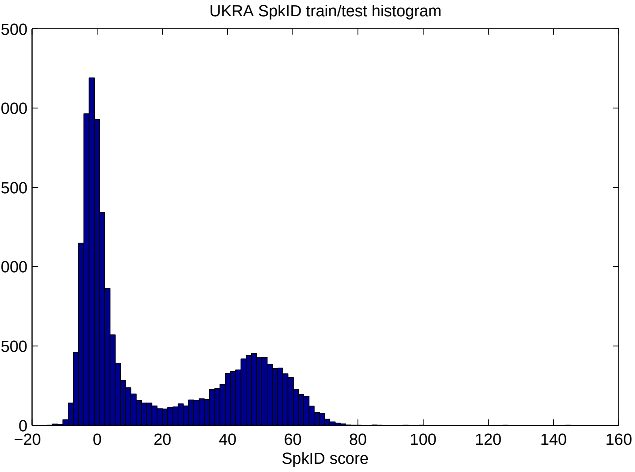

Figure 3.1: Histogram of speaker ID scores – example for Ukrainian.

test segments. Next, we added a channel compensation: The system is the same, except that we use channel-compensated first-order statistics everywhere.

We evaluated the systems on the 14 languages of the closed-set language detection task of the NIST 2007 Language Recognition Evaluation (LRE 2007), with input segments of nominal duration 30 seconds [18]. Our development data, used to train all system parameters, was the same data we used in preparation for the NIST LRE 2007 evaluations and does not overlap with the LRE 2007 evaluation data, see [6]. We used C^*_{avg} as evaluation criterion, see [27].

The results are $\mathbf{C^*_{avg} = 11.32}$ for the system without the channel compensation and $\mathbf{C^*_{avg} = 1.74}$ for the system with the channel compensation applied. These results show, that the channel compensation gives dramatic reduction in error-rate and that GMM factor-analysis modeling can be used to build accurate acoustic language recognizers. We expect an improvement also for shorter duration segments and we plan to experimentally prove this expectation.

3.6 Experiments with PIVCO

The results are reported for the NIST LRE 2007 primary condition, for three tasks reflecting the nominal length of the testing utterances – 30, 10 and 3 seconds. As the metric, $100 \times C_{avg}$ (see [18] for formulas) is used. All results are presented for calibrated systems using linear backed (LDA) followed by linear logistic regression [29] (LLR).

The phonotactic systems are based on hybrid ANN/HMM approach, where artificial neural networks (ANN) are used to estimate posterior probabilities of phonemes from Mel filter bank log energies using the context of 310ms around the current frame. Hybrid recognizer is trained for Hungarian on the SpeechDat-E databases. For more details see [15, 16].

Table 3.8 describes influence of LFA to the phonotactic system with Binary decision trees. It mainly helps for 30 second condition. We observed little or no improvement in case of 10 and 3 seconds tasks, where little data for model adaptation was available.

The results indicate, that using the factor analysis for the inter-session variability compensation in phonetic recognition followed by language model (PRLM) improves the performance in the LRE systems for 30 second condition.

Table 3.8: Binary decision trees with PIVCO ($100 \times C_{avg}$)

LRE 2007	30	10	3
HU_Tree	5.58	11.54	23.45
HU_Tree_PIVCO	5.01	11.45	23.83

3.7 Experiments with the i-vector Based System

We have built several classifiers using 400 dimensional i-vectors as feature vectors. The results are reported on the *30 second* closed-set condition of the NIST LRE 2009 task. In all experiments, we have done the calibration, which was trained on our LRE2009 DEV set. The back-end was a Gaussian model followed by a Discriminative Linear Regression model [27].

3.7.1 Amount of Data for Training the i-vector Extractor

In the beginning we had run a series of experiments to assess the influence of the amount of the training data for training the i-vector extractor. Results indicate, that it is beneficial to use as much data as possible. For this experiment, we used a ML-based classifier described in the following section. We observed a drop in the accuracy when using a reduced database with 500 utterances[3] per language for training. Performance had dropped from $C_{avg} = 1.85\%$ when training the i-vector extractor on the whole TRAIN database to $C_{avg} = 1.99\%$ when using the reduced database.

3.7.2 ML-based i-vector System

The best performing classifier is based on a Maximum Likelihood approach. For each language, we have built a generative Gaussian model, that was trained on all available data for the particular language. To obtain a score of the test utterance, we compute a log-likelihood over each language model. Using this simple approach, we were able to obtain already mentioned performance of $C_{avg} = 1.85\%$.

We have also tried to improve this method by applying the dimensionality reduction and normalization techniques. When applying Linear Discriminant Analysis (LDA), which allows us to reduce the dimension of the i-vector to just 22 dimensions, we obtained a result of $C_{avg} = 1.92\%$. The benefit of this approach is an extremely compact representation of the i-vector and speed of such system for the price of a small deterioration of the accuracy. In the following experiment, we tried to normalize all i-vectors to the unit length, which effectively forces them to be Gaussian-distributed. Unfortunately, applying the length normalization did not bring any improvement as the performance dropped to $C_{avg} = 1.97\%$.

3.7.3 PLDA-based i-vector System

The second generative model employs Probabilistic Linear Discriminant Analysis (PLDA) [30]. This technique models an inter-language space by the across-class covariance matrix of the i-vectors of all languages and an intra-language space by the within-class covariance matrix of the languages. The performance of this system was $C_{avg} = 1.97\%$.

3.7.4 SVM-based and LR-based i-vector Systems

The third system employs a discriminative approach based on Support Vector Machines (SVM), where the i-vectors are used directly as features. In this case, we obtain $C_{avg} = 2.14\%$. Applying Nuisance Attribute Projection (NAP) or LDA did not bring any improvement.

[3] For the languages with less than 500 utterances we take all of the data available

The fourth system is based on discriminative L2-regularized Linear Regression (LR). With this system, we obtain a $C_{avg} = 2.05\%$. However in this case, by applying LDA we improve the performance to a $C_{avg} = 1.91\%$ and by applying NAP we obtain a result of $C_{avg} = 1.93\%$.

Chapter 4

Conclusions

We introduced a simple but promising approach of acquiring telephone data for LID. Experiments with selected languages using standard telephone data and telephone data acquired from broadcast were performed. Both phonotactic and acoustic approaches for recognition were investigated.

Obtained results show, that if systems trained on broadcast data are used to recognize CTS, the performance is significantly lower than it would be with the systems trained on target data. However, experiments with channel compensation techniques indicate, there is a possibility to improve the performance by investigating other compensation techniques to suppress the distortion caused by passing the telephone call through wide-band channel. On the other hand, training the systems on CTS data and testing on broadcast data seems to be all right as the same trends are observed for the CTS-based test sets.

Performed experiments show, that if broadcast data are used both for training and testing, the performance is excellent but if the CTS data are used to evaluate the system, the performances drops dramatically. This is probably because the systems trained and tested on broadcast data have learned some information about the channel of particular broadcast, especially if all samples of the same language come from one radio station, but this problem deserves further investigation. As soon a database exists, where one language comes from different broadcasts, experiments should be conducted to verify this assumption.

Results of the experiments also lead to a claim, that the broadcast data are "easier", as they contain mostly clean, prepared and grammatically correct speech. This idea is supported by the fact, that broadcast data were always (except the case when the channel compensation trained on broadcasts was used) recognized by systems trained on CTS data with better accuracy than NIST 2007 data containing a lot of unclean speech.

It should be stressed, that the results of systems trained on broadcasts were obtained on automatically created databases without human annotator checking and several compromises were made, especially considering Farsi language as Dari and using French spoken in the African region. Also, only *6 hours* of training data per language was used to train systems on broadcast data in comparison with average *28 hours* of training data per language for systems trained on CTS data.

The performance of the systems trained on broadcast data simulates a scenario, when no standard CTS training data are available and we need to detect a particular language. Although the results are significantly worse than ones we would get with CTS data for training, using the broadcast data can be the only option in such situation.

According to the updated work-plan for the second phase of the project, we have concentrated on the issues of repeating speakers in the broadcast data and we have obtained substantial improvements of performances by filtering these repeating speakers with a speaker identification system.

Also, we have successfully applied several advanced channel compensation techniques with excellent results on NIST-defined LRE 2007 and 2009 task.

We have worked on inter-session variability compensation in an acoustic system, by applying restricted joint-factor analysis (JFA) technique originally investigated for speaker recognition. This led to substantial improvement in accuracies.

Next, we have extended the intersession compensation also to the phonotactic approach by suggesting and performing experiments with “PIVCO”, with encouraging results.

Finally, we have adapted the best performing technique from the current state-of-the art speaker recognition systems by using i-vectors, which are a compact low-dimensional representation of each utterance in a total variability space. This representation allowed us to build very small language recognition systems using basic classification techniques. Obtained results are very competitive with the latest state-of-the art LRE systems.

This work also helped NIST and LDC to analyze and use new sources of the data for building LRE systems, which were later extensively used in the NIST LRE 2009 evaluations. Using the broadcast data has allowed NIST and LDC to use the largest number of languages to date, while keeping the costs of creating such database within reasonable boundaries.

The results of participants and the subsequent discussion during the workshop have revealed, that using the broadcast data is beneficial for building large-scale LRE systems.

Appendix A

Detailed Results

Table A.1: Results of phonotactic system based on string output. System trained on CallFriend database.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	1.517	3.858	10.476
English	2.529	10.682	3.761
French	2.440	2.237	10.285
Hindi	2.142	10.202	6.238
Korean	0.208	6.142	5.000
Mandarin	0.952	11.166	4.000
Spanish	0.714	4.343	1.523
Vietnamese	0.684	7.314	5.619
Average	1.398	6.993	5.863
pooled minDET	1.700	7.627	6.261
pooled EER	1.781	7.736	6.583
pooled unweighted minDET	1.794	9.072	6.261
pooled unweighted EER	1.982	9.122	6.583

Table A.2: Results of phonotactic system based on string output. System trained on broadcast database.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	19.077	19.371	0.7142
English	11.666	19.698	2.666
French	13.839	20.968	1.285
Hindi	17.440	21.619	0.904
Korean	6.994	12.737	1.142
Mandarin	9.017	22.321	0.238
Spanish	5.000	10.200	0.666
Vietnamese	4.970	12.335	0.285
Average	11.000	17.406	0.988
pooled minDET	11.644	18.286	1.333
pooled EER	11.949	18.593	1.416
pooled unweighted minDET	12.035	18.796	1.333
pooled unweighted EER	12.250	19.122	1.416

Table A.3: Results of phonotactic system based on lattices. System trained on CallFriend database.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	0.744	3.707	6.714
English	1.726	11.108	3.761
French	0.803	1.976	7.761
Hindi	0.446	8.609	5.523
Korean	0.208	5.720	3.571
Mandarin	0.535	8.762	2.142
Spanish	0.148	3.904	1.857
Vietnamese	0.625	5.977	4.428
Average	0.654	6.221	4.470
pooled minDET	0.822	6.903	5.083
pooled EER	0.900	6.995	5.232
pooled unweighted minDET	0.866	7.769	5.083
pooled unweighted EER	0.875	7.836	5.232

Table A.4: Results of phonotactic system based on lattices. System trained on broadcast database.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	14.791	18.498	0.428
English	10.029	17.637	1.904
French	11.339	18.696	1.428
Hindi	12.559	16.958	0.666
Korean	3.839	8.172	1.142
Mandarin	5.446	16.762	0.333
Spanish	2.142	8.616	0.904
Vietnamese	2.886	9.717	0.047
Average	7.879	14.382	0.857
pooled minDET	8.697	15.017	1.220
pooled EER	8.958	15.215	1.398
pooled unweighted minDET	9.258	15.313	1.220
pooled unweighted EER	9.607	15.497	1.398

Table A.5: Results of acoustic system trained on CallFriend database without channel compensation.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	2.083	2.359	5.190
English	1.488	13.291	10.619
French	4.077	2.531	12.333
Hindi	4.255	15.340	13.238
Korean	2.291	7.788	6.238
Mandarin	2.559	10.153	5.666
Spanish	4.315	8.564	2.761
Vietnamese	1.160	3.661	4.190
Average	2.779	7.961	7.529
pooled minDET	3.277	8.670	8.184
pooled EER	3.407	8.807	8.261
pooled unweighted minDET	3.276	10.601	8.184
pooled unweighted EER	3.375	10.873	8.261

Table A.6: Results of acoustic system trained on broadcast database without channel compensation.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	21.577	23.920	1.142
English	9.375	22.564	3.761
French	16.220	20.010	3.666
Hindi	20.952	24.347	2.476
Korean	11.428	15.325	3.190
Mandarin	9.017	20.200	0.714
Spanish	10.000	14.083	1.428
Vietnamese	7.351	7.847	1.857
Average	13.240	18.537	2.279
pooled minDET	13.958	19.317	2.833
pooled EER	14.423	19.502	3.250
pooled unweighted minDET	13.357	20.133	2.833
pooled unweighted EER	13.633	20.350	3.250

Table A.7: Results of acoustic system trained on CallFriend database with channel compensation trained on broadcast data.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	0.625	1.432	11.428
English	1.101	9.014	8.714
French	1.250	0.949	12.333
Hindi	0.654	10.878	7.666
Korean	0.148	4.563	3.238
Mandarin	0.803	7.139	5.619
Spanish	1.011	3.580	2.523
Vietnamese	0.148	3.556	9.761
Average	0.718	5.139	7.660
pooled minDET	1.104	5.447	8.166
pooled EER	1.145	5.644	8.250
pooled unweighted minDET	1.196	6.746	8.166
pooled unweighted EER	1.250	6.959	8.250

Table A.8: Results of acoustic system trained on broadcast database with channel compensation trained on broadcast data.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	15.565	17.015	0.142
English	6.517	15.538	0.476
French	11.458	13.324	0.238
Hindi	14.851	17.612	0.000
Korean	6.994	10.583	0.809
Mandarin	6.666	16.875	0.000
Spanish	6.428	10.113	0.714
Vietnamese	2.678	9.773	0.142
Average	8.895	13.854	0.315
pooled minDET	9.471	14.510	0.505
pooled EER	9.840	15.013	0.583
pooled unweighted minDET	9.196	15.939	0.505
pooled unweighted EER	9.508	16.198	0.583

Table A.9: Results of acoustic system trained on CallFriend database with channel compensation trained on telephone data.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	0.178	1.471	3.190
English	0.416	6.850	4.619
French	0.446	0.587	4.619
Hindi	0.178	7.643	2.190
Korean	0.208	2.923	1.285
Mandarin	0.505	8.168	1.285
Spanish	0.119	2.636	0.380
Vietnamese	0.000	2.005	1.142
Average	0.256	4.035	2.339
pooled minDET	0.383	4.258	2.964
pooled EER	0.420	4.296	3.083
pooled unweighted minDET	0.437	5.714	2.964
pooled unweighted EER	0.500	5.730	3.083

Table A.10: Results of acoustic system trained on broadcast database with channel compensation trained on telephone data.

Language	**NIST 2003**	**NIST 2007**	**Broadcast**
Dari	11.220	17.892	0.047
English	7.410	17.296	1.666
French	13.839	13.585	0.238
Hindi	14.970	15.280	0.095
Korean	3.720	6.250	0.714
Mandarin	9.166	20.584	0.000
Spanish	6.726	9.027	0.619
Vietnamese	2.023	5.336	0.000
Average	8.634	13.156	0.422
pooled minDET	9.136	14.136	0.886
pooled EER	9.222	14.290	0.922
pooled unweighted minDET	8.964	15.772	0.886
pooled unweighted EER	9.107	15.860	0.922

Appendix B

Figures

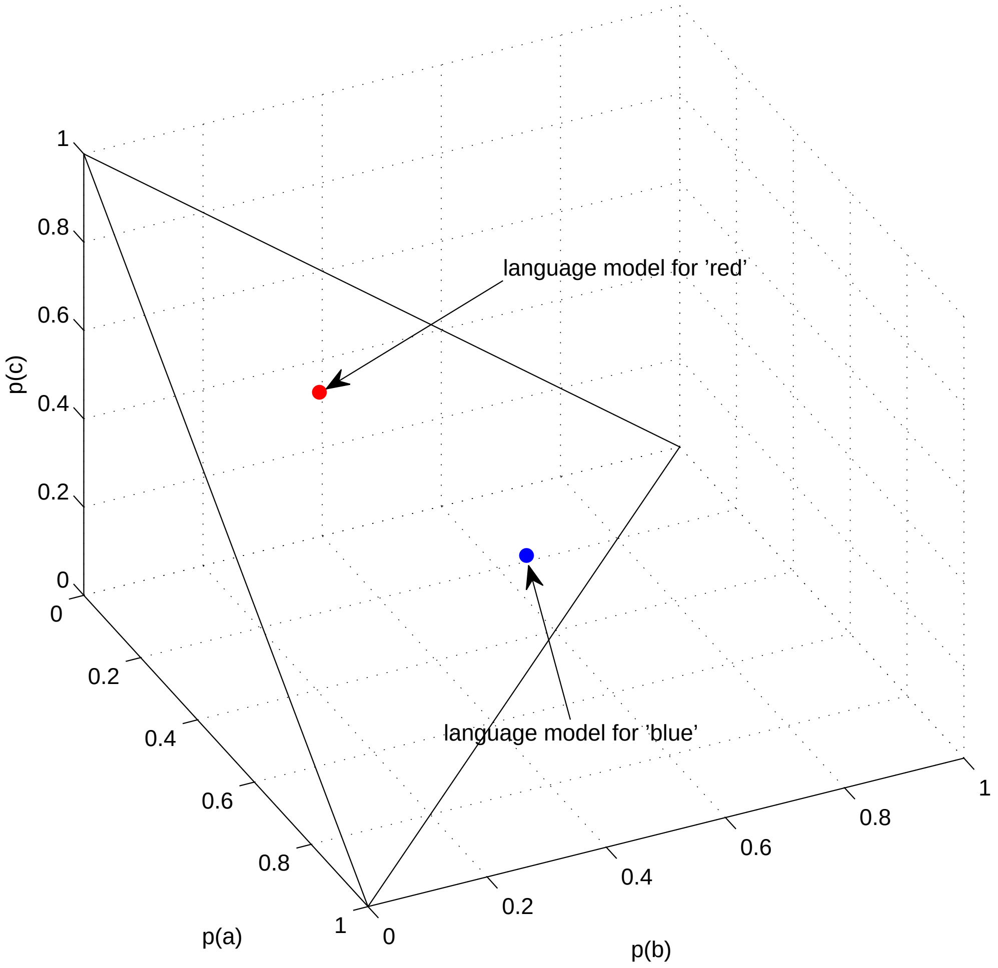

Figure B.1: Simplex with two language models

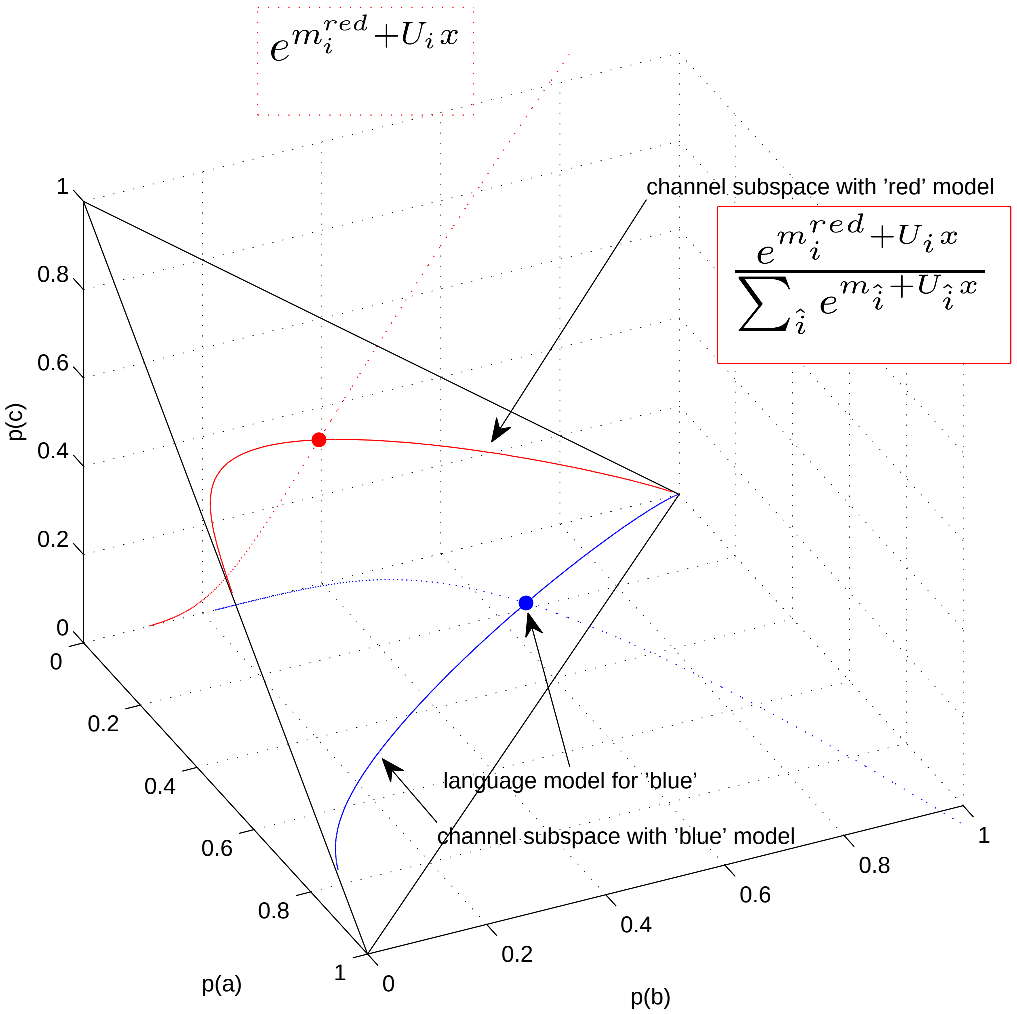

Figure B.2: Simplex with two language models. Red and blue lines show the subspace defining the inter-session variability.

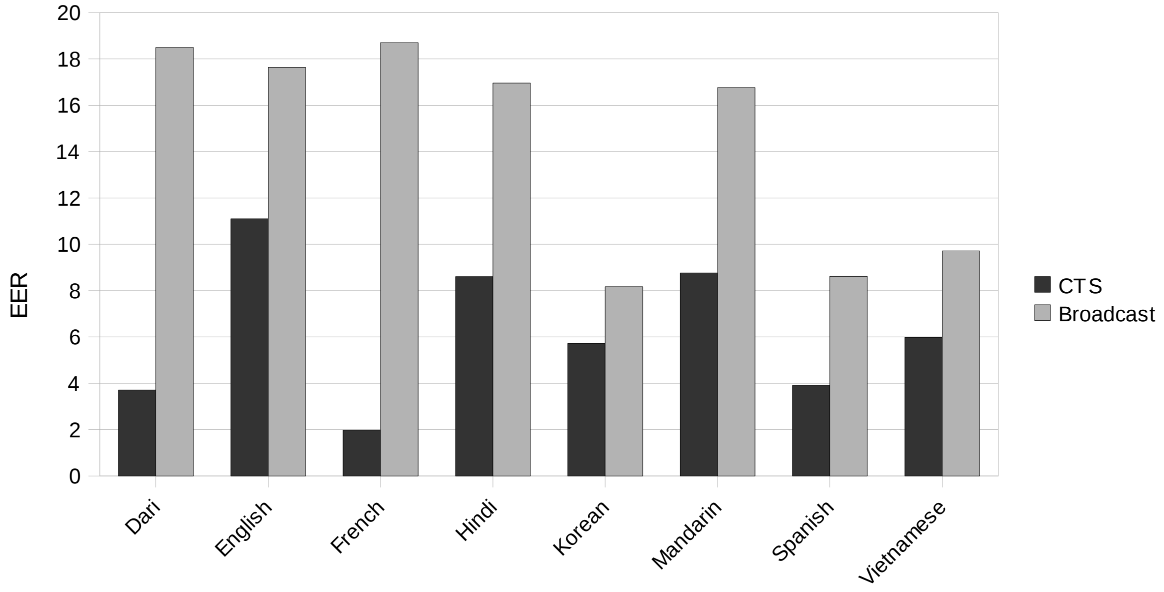

Figure B.3: Equal Error Rate of individual languages for phonotactic systems based on lattices trained on CTS and broadcasts.

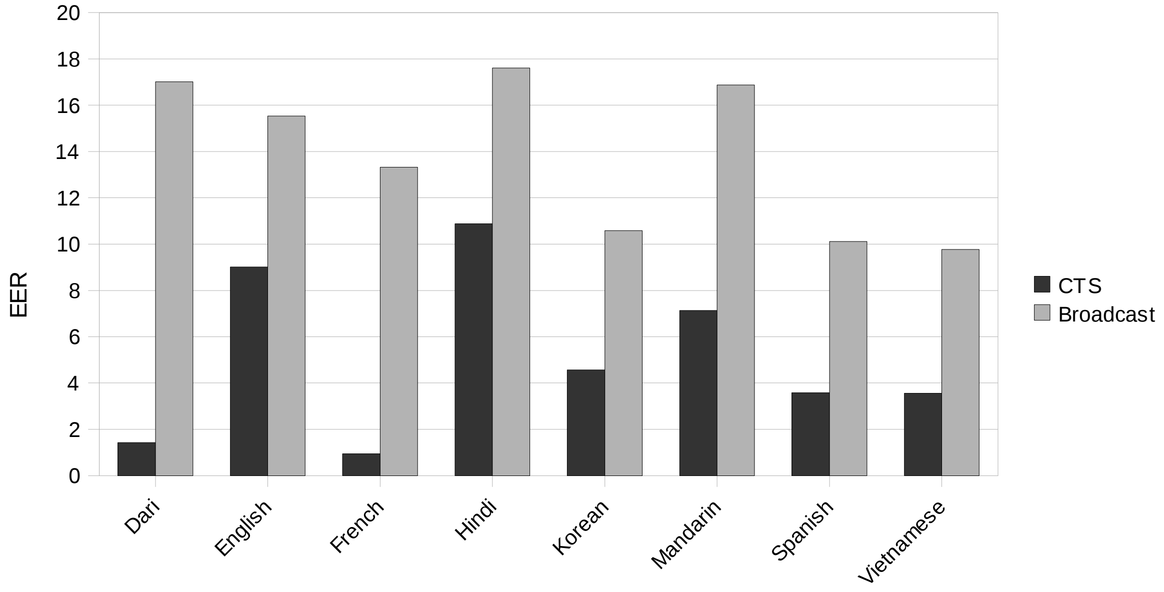

Figure B.4: Equal Error Rate of individual languages for acoustic systems trained on CTS and broadcasts with channel compensation trained on broadcast data.

Appendix C

Cooperation with Linguistic Data Consortium

We were collaborating with the Linguistic Data Consortium (LDC) on preparation of broadcast data database, which will contain recording from various radio stations in many languages. Language labels of all recordings in this database need to be manually verified. Verification of such large amount of data consisting of tens of languages represents a problem in routing a recordings to an annotator, able to recognize language of particular recording.

We received a set of various broadcast recordings from LDC without language labels. It was expected, that these recordings contain 39 different languages. [1] This package contained over 7GB or 10150 files of stereo recordings compressed in mp3 format. Given the fact, that the recordings often contain different broadcast stations in the left and right channel, more than 14000 hours of data had to be processed and labeled.

In order to label the data, we downloaded large amount of broadcast data from the Voice of America archive, where the recordings **are labeled** according to location of broadcasting and predominant language. We prepared the data for training using the same techniques explained in section 2.6.1 and trained a phonotactic system based on string output from our Hungarian phoneme recognizer. The language models were trained for 43 languages [2] and there was an average of 14.1 hours of speech per each language for training. However, this number varied from 4.7 hours (for Serbian)to 64 hours (for Korean).

We provided three top-scoring language labels for each file and each channel to speed up the routing of files to human annotators. We also provided speech and non-speech labels and labels for the phone calls detected in the broadcasts. These labels were obtained by techniques explained in sections 2.1, 2.2 and 3.

We have also created software packages for phone call detection and speech/non-speech segmentation. This software was shipped to LDC will allow them to process the recorded broadcast more effectively.

[1] Albanian, Amharic, Armenian, Azeri, Bengali, Bosnian, Burmese, Cantonese, Creole, Croatian, Dari, English, French, Georgian, Greek, Hausa, Hindi, Indonesian, Khmer, Korean, Kurdish, Lao, Mandarin, Pashto, Persian, Portuguese, Russian, Serbian, Shona, Somali, Spanish, Swahili, Thai, Tigrigna, Turkish, Ukrainian, Urdu, Uzbek, Vietnamese

[2] Albanian, Amharic, Azerbaijani, Bengali, Bosnian, Burmese, Cantonese, Creole, Croatian, Dari (Persian), English, French, Georgian, Greek, Hausa, Hindi, Indonesian, Khmer, Kinyarwanda, Korean, Kurdish, Lao, Macedonian, Mandarin, Ndebele, Oromo, Pashto, Persian, Portuguese, Russian, Serbian, Shona, Somali, Spanish, Swahili, Thai, Tibetan, Tigrinya, "Talk To America - English", Turkish, Ukrainian, Urdu, Uzbek, Vietnamese

Bibliography

[1] Ivo Řezníček, “Audiovisual recording system,” Diploma thesis, Brno University of Technology FIT, 2007.

[2] N. Brummer et al, “Discriminative acoustic language recognition via channel-compensated gmm statistics,” in *Proc. Interspeech*, 2009.

[3] P. Ouellet P. Kenny, G. Boulianne and P. Dumouchel, “Joint factor analysis versus eigenchannels in speaker recognition,” *IEEE Transactions on Audio, Speech and Language Processing 15 (4)*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[4] P. Kenny, N. Dehak, P. Ouellet, V. Gupta, and P. Dumouchel, “Development of the primary crim system for the nist 2008 speaker recognition evaluation,” in *Proceedings of Interspeech 2008*, sep 2008.

[5] Ondřej Glembek, Lukáš Burget, Najim Dehak, Niko Brummer, and Patrick Kenny, “Comparison of scoring methods used in speaker recognition with joint factor analysis,” in *Proc. ICASSP*, 2009.

[6] P. Matějka V. Hubeika, L. Burget and P. Schwarz, “Discriminative training and channel compensation for acoustic language recognition,” in *Proc. Interspeech*, 2008.

[7] P. Matějka, L. Burget, P. Schwarz, and J. Černocký, “Brno university of technology system for nist 2005 language recognition evaluation,” in *Proc. NIST LRE 2005 Workshop*, San Juan, Puerto Rico, June 2006, pp. 57–64.

[8] Ondřej Glembek, Pavel Matějka, Lukáš Burget, and Tomáš Mikolov, “Advances in phonotactic language recognition,” in *Proc. Interspeech 2008*. 2008, p. 4, International Speech Communication Association.

[9] J. Navratil, “Spoken language recognition - a step toward multilinguality in speech processing,” *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 6, pp. 678–685, September 2001.

[10] W. Andrews J. Navrátil, Q. Jin and J.P. Campbell, “Phonetic speaker recognition using maximum-likelihood binary-decision tree models,” in *Proc. of ICASSP*, April 2003.

[11] P. Kenny, “Joint factor analysis of speaker and session variability : Theory and algorithms - technical report CRIM-06/08-13. Montreal, CRIM, 2005,” 2005.

[12] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. PP, no. 99, 2010.

[13] P. Kenny, G. Boulianne, P. Oullet, and P. Dumouchel, “Joint factor analysis versus eigenchannes in speaker recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[14] Niko Brümmer, "The EM algorithm and minimum divergence," Agnitio Labs Technical Report. Online: http://niko.brummer.googlepages.com/EMandMINDIV.pdf, Oct. 2009.

[15] Petr Schwarz, Pavel Matějka, and Jan Černocký, "Hierarchical structures of neural networks for phoneme recognition," in *Proceedings of ICASSP 2006*, 2006, pp. 325–328.

[16] Petr Schwarz, Pavel Matějka, and Jan Černocký, "Towards lower error rates in phoneme recognition," in *Proceedings of 7th International Conference Text,Speech and Dialoque*, 2004.

[17] "The 2003 NIST Language Recognition Evaluation Plan (LRE03)," http://www.nist.gov/speech/tests/lre/2003/LRE03EvalPlan-v1.pdf.

[18] "The 2007 NIST Language Recognition Evaluation Plan (LRE07)," http://www.nist.gov/speech/tests/lang/2007/LRE07EvalPlan-v8b.pdf.

[19] Pavel Matějka, Lukáš Burget, Ondřej Glembek, Petr Schwarz, Valiantsina Hubeika, Michal Fapšo, Tomáš Mikolov, and Oldřich Plchot, "But system description for nist lre 2007," in *Proc. 2007 NIST Language Recognition Evaluation Workshop*. 2007, pp. 1–5, National Institute of Standards and Technology.

[20] Niko Brummer and David van Leeuwen, "On calibration of language recognition scores," in *Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006, pp. 1–8.

[21] Lukáš Burget, Pavel Matějka, Petr Schwarz, Ondřej Glembek, and Jan Černocký, "Analysis of feature extraction and channel compensation in GMM speaker recognition system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 1979–1986, 2007.

[22] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, jan 2000.

[23] Pedro A. Torres-Carrasquillo, Elliot Singer, Mary A. Kohler, Richard J. Greene, Douglas A. Reynolds, J.R. Deller, and Jr., "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. 7 th International Conference on Spoken Language Processing*, Denver, Colorado, USA, Sept. 2002.

[24] Jordan Cohen, Terri Kamm, and Andreas G. Andreou, "Vocal tract normalization in speech recognition: Compensating for systematic speaker variability," *The Journal of the Acoustical Society of America*, vol. 97, no. 5, pp. 3246–3247, 1995.

[25] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proc. Eurospeech*, Rhodes, Greece, Sept. 1997, pp. 963–966.

[26] Niko Brummer, "Spescom DataVoice NIST 2004 system description," in *Proc. NIST Speaker Recognition Evaluation 2004*, Toledo, Spain, June 2004.

[27] N. Brümmer, L. Burget, O. Glembek, V. Hubeika, Z. Jančík, M. Karafiát, P. Matějka, T. Mikolov, O. Plchot, and A. Strasheim, "But system description for nist lre 2009," in *Proc. NIST Language recognition workshop 2009*.

[28] P. Matejka, L. Burget, P. Schwarz, O. Glembek, M. Karafiat, F. Grezl, J. Cernocky, D. van Leeuwen, N. Brummer, and A. Strasheim, "Stbu system for the nist 2006 speaker recognition evaluation," in *Proc of ICASSP*, 2007, pp. 221–224.

[29] Niko Brümmer, Lukáš Burget, Jan Černocký, Ondřej Glembek, František Grézl, Martin Karafiát, David Leeuwen van, Pavel Matějka, Petr Schwarz, and Albert Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[30] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *11th International Conference on Computer Vision*, 2007, pp. 1–8.

Appendix D

List of Abbreviations

ANN	Artificial Neural Network
BUT	Brno University of Technology
CTS	Continuous Telephone Speech
DCF	Decision Cost Function
DET	Detection Error Trade-off
DVB	Digital Video Broadcasting
DVB-C	Digital Video Broadcasting - Cable
DVB-S	Digital Video Broadcasting - Satellite
DVB-T	Digital Video Broadcasting - Terrestrial
EER	Equal Error Rate
ELLR	Expected Log Likelihood Ratio
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
JFA	Joint Factor Analysis
LDA	Linear Discriminant Analysis
LDC	Linguistic Data Consortium
LID	Language Identification
LRE	Language Recognition
MAP	Maximum A-posteriori Probability
MFCC	Mel-frequency Cepstral Coefficients
ML	Maximum Likelihood
MP3	MPEG-1 Layer III
MPEG	Motion Picture Experts Group
NIST	National Institute of Standards and Technology
PIVCO	Phonotactic Inter-Session Variability Compensation
PLDA	Probabilistic Linear Discriminant Analysis
PSD	Power Spectral Density
SDC	Shifted Delta Cepstra
SRE	Speaker Recognition
SVM	Support Vector Machines
UBM	Universal Background Model
VTLN	Vocal Tract Length Normalization
VoA	Voice of America